## REMARKS

After entry of the above amendments, claims 1-27 will be pending in the above-identified application. Claims 13 and 21 have been amended to correct clerical errors and make explicit what was inherent. New claim 27 is supported in the specification. No new matter has been added.

### Claims 1-12

Claims 1-12 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,363,388 (hereinafter "Sprenger") in view of U.S. Patent No. 6,157,706 (hereinafter "Rachelson").

The Office action states:

> Sprenger discloses providing said data of an information record to a consumer, (Col. 28, lines 20-28 and Col. 25, lines 52-58); and updating a history table, said history table comprising a message table field, (Col. 13, lines 50-64 and Col. 22, lines 51-68).

(Pg. 6 of October 9, 2002 Office action).

The first Sprenger passage cited as disclosing updating a history table discloses:

> The ArchiverAgent hosts the ArchiverMinion, which supports the client and project multi-step archiving procedure. The ArchiverMinion will be configurable from the command line to perform a variety of archiving tasks, each of which may be scheduled with the job management system to run at regular intervals that vary with the type of archiving functions being performed. The archiving process may require operator intervention because physical media such as tapes and disks must generally be moved between racks and drives. The archiving procedure consists of the following general steps: unloading relevant data from the database; updating selected tables to reflect that the client or project has been archived; notifying operators to move data to tape; and removing archived files.

(Col. 13, ll. 50-64).

The second Sprenger passage cited as disclosing updating a history table discloses:

> CustTarget 434 is an association table between Customer 400 and BrandCategory 432 that can store which particular segment a customer is in for this brand category. CustTarget 434 may also contain volume information that can be later used to segment (or re-

> segment) the customer. CustTargetHst 435 is a history table for CustTarget 434 that can store the historical segment and volume information for a customer.
>
> Brand 412 is a grouping table that names the brands being tracked on behalf of the client. Note that competitive brands may be included.
>
> CustBrandAssoc 436 is an association table between Customer 400 and Brand 412. CustBrandAssoc 436 stores data about the relationship a customer has with a given brand. For example, "Out of the last 10 times you have used a product in this brand category, how many were for brand X"? CustBrandHst 437 is a history table for CustBrandAssoc 436 that can store the historical relationships a customer has with a given brand.

(Col. 22, l. 51 to col. 23, l. 2).

Sprenger discloses "a history table . . . that can store historical segment and volume information for a customer" and "a history table . . . that can store the historical relationships a customer has with a given brand." (Col. 22, ll. 56-58; col. 22, l. 67 to col. 28, l. 2). In contrast, claim 1 recites a "history table comprising one or more history records, each said history record comprising a message state field." Furthermore, claim 1 recites "updating a history table, . . ., said updating comprising setting said message state field in a history record corresponding to said consumer to indicate said consumer accessed said data." Therefore, Sprenger does not disclose "updating a history table, said history table comprising one or more history records, each said history record comprising a message state field, said updating comprising setting said message state field in a history record corresponding to said consumer to indicate said consumer accessed said data" as recited in claim 1.

The Office action further states:

> Sprenger does not clearly teach, "said updating comprising setting said message state field in a history record corresponding to said consumer to indicate said consumer accessed said data."
>
> However, Rachelson shows said updating comprising setting said message state field in a history record corresponding to said consumer to indicate said consumer accessed said data, (Col. 9, lines 8-35).

(Pg. 6 of October 9, 2002 Office action).

The cited passage of Rachelson discloses:

> FIG. 9(b) shows an example data structure for storing messages received by a user. Each message is assigned a message id and has one or more flags indicating whether the message has been read by the user and whether the user has archived the message.
>
> If the user selects "List past messages," the user is prompted to determine which past messages the user wishes to view (by month and year) and the system prepares pages containing a printed list (sender, date, messages id, subject, and number of pages) of the messages specified by the user. Control then passes to step C of FIG. 6, where the prepared pages are faxed to the user at the specified fax machine. A preferred embodiment of the present invention also allows the user to retrieve listings of past messages sent/received from a specific person in the user's address book.
>
> If the user selects "Retrieve past messages," the user is prompted to specify which past messages the user wishes to view (by message id) and the system prepares pages containing of the contents of the specified messages. Control then passes to step C of FIG. 6, where the prepared pages are faxed to the user at the specified fax machine.
>
> If the user selects "Forward past messages to someone in address book," the user is prompted to specify which past messages the user wishes to send and to which internet fax number he wishes to send them. The system then prepares pages containing of the contents of the specified messages and the prepared pages are e-mailed to the specified recipient.

(Col. 9, ll. 8-35).

Rachelson discloses "[e]ach message is assigned a message id and has one or more flags indicating whether the message has been read by the user and whether the user has archived the message." (Col. 9, ll. 9-12). In contrast, claim 1 recites a "history table comprising one or more history records, each said history record comprising a message state field." Moreover, claim 1 recites "updating a history table, . . ., said updating comprising setting said message state field in a history record corresponding to said consumer to indicate said consumer accessed said data." Hence, Rachelson does not disclose "updating a history table, said history table comprising one or more history records, each said history record comprising a message state field, said updating comprising setting said message state field in a history record corresponding to said consumer to indicate said consumer accessed said data" as recited in claim 1.

Even if Sprenger and Rachelson were combined, the combination would neither teach nor suggest "updating a history table, said history table comprising one or more history records, each said history record comprising a message state field, said updating comprising setting said message state field in a history record

corresponding to said consumer to indicate said consumer accessed said data" as recited in claim 1. Therefore, Applicants respectfully submit that claim 1 is patentable over Sprenger in view of Rachelson. Given that claims 2-12 depend from claim 1, it is respectfully submitted that those claims are also patentable over Sprenger in view of Rachelson.

Claims 13-20 and 27

Claims 13-20 have been rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,058,389 (hereinafter "Chandra").

The Office action states:

> Chandra et al. disclose a system for the delivery of information to multiple consumers, said system comprising: an information queue comprising one or more information queue records, each said information queue record comprising information to be accessed by one or more consumer, (Col. 6, lines 45 – Col. 8, lines 60 and see fig. 3); a table comprising one ore more table records, each said table record comprising an information identification field comprising an identification of said information in an information queue record, each said table record further comprising a consumer identification field comprising an identification of one or said one or more consumers, (Col. 6, lines 45 – Col. 8, lines 60 and see fig. 3).

(Pgs. 3-4 of October 9, 2002 Office action).

The cited passage of Chandra discloses:

> Database files 322, 324 of the relational database system 300 also store queue tables comprising queues and messages of queues, as shown in FIG. 2. The units processed by the queuing system are messages 208. A message 208 represents the smallest unit of work processed by a single transaction, for example, a request for processing by an application. A transaction can create or process multiple messages. A message comprises user data and control information, also called payload and metadata, respectively. The user data of a message is passed form the process or application program which created the message to another application, transaction or process, based on what is in the control information, but without modification of the user data. The control information represents message properties used by the queuing system to manage messages. Transactions can create messages using an ENQUEUE operation and consume messages by using a DEQUEUE operation. Messages are selected for consumption based upon the control information stored with the message.
>
> A queue 202, 204 represents a collection of messages 208 ordered as in a list. Each message belongs to only one queue. Queues 202, 204 can be created, altered, started, stopped

and dropped using the operations and processes described herein. Each queue 202, 204 is represented by a KGL object, and is referenced by an object name (the queue name) and its owner name (the schema name) in the library cache.

A queue table 200 holds a set of queues, and each queue table 200 is implemented as a database table within the relational database system 300. Each queue table 200 can contain multiple queues 202, 204 each having multiple queue messages 208. In one embodiment, a queue table 200 is structured as shown in FIG. 2. Each row of the queue table 200 represents a message 208 in a queue 202, 204. Some of the columns in a queue table 200 are meta-data describing a queue 202, 204. In one embodiment, each row of the queue table 200 has the following columns:

| Column | Contents |
|---|---|
| QUEUE | Queue name |
| MSG_ID | Message identifier |
| CORR_ID | User-provided correlation identifier |
| MSG_PRIORITY | Message priority |
| MSG_STATE | State of the message (READY to be processed, DELAYED, PROCESSED, or EXPIRED |
| DELAY | Time after which the message will be ready to be processed |
| EXPIRATION | Message expiration time in seconds |
| TIME_MANAGER_INFO | Date used by time manager process to monitor messages |
| LOCAL_ORDER_NO | Local order number of messages |
| CHAIN_NO | Chain number of message |
| DSCN | Dependent transaction number |
| CSCN | Commit transaction number |
| ENQ_TIME | Original enqueue time |
| ENQ_USER_ID | User identifier for the user who enqueued the message |
| ENQ_TXN_ID | Current transaction identifier |
| DEQ_TIME | Time when the message was dequeued |
| DEQ_TXN_ID | Transaction which performed the dequeue |
| DEQ_USER_ID | User id of the user who dequeued the message |
| RETRY_COUNT | Number of retries |
| REPLY_QUEUE_OWNER | Reply queue schema |
| REPLY_QUEUE | Reply queue name |
| EXCEPTION_QUEUE_OWNER | Exception queue schema |
| EXCEPTION_QUEUE | Exception queue name |
| ARG_COUNT | Number of arguments |
| USER_DATA | User data for use by an application or process |

Configuration information for queue tables and queues are stored in two dictionary tables. A Queue Table Dictionary Table 212 stores configuration information for queue tables and a Queue Dictionary Table 210 stores configuration information for all queues defined in the system. In one embodiment, each row of the Queue Table Dictionary Table

212 has configuration information for one queue table 200, and the Queue Table Dictionary 212 has the following columns:

| Column | Contents |
|---|---|
| OWNER | Queue table schema |
| QUEUETABLE | Queue table name |
| TYPE | Type of user data stored in queues in the queue table (OBJECT TYPE for user-defined object types or VARIANT for internal use |
| OBJECT_TYPE | Abstract data type of user data stored in queues of the queue table (provided only when TYPE has the value OBJECT TYPE) |
| OBJECT_NUMBER | Number of queue table object |
| TABLE_COMMENT | User comment about the queue table |

In this embodiment, each row of the Queue Dictionary Table 210 represents a queue defined in a queue table of the queuing system, and the Queue Dictionary Table 210 has the following columns:

| Column | Contents |
|---|---|
| OWNER | Queue schema name |
| NAME | Queue name |
| QUEUETABLE | Queue table containing the queue named NAME |
| QID | Identifier for the queue |
| QTYPE | Queue type (NORM for normal queue, EXPT for exception queue) |
| MAX_RETRIES | Number of times a message is processed before being moved to an exception queue |
| RETRY_INTERVAL | Time lapse before retry takes place |
| ENQUEUE | Boolean flag identifying whether ENQUEUE is disabled/enabled |
| DEQUEUE | Boolean flag identifying whether DEQUEUE is disabled/enabled |
| TRACKING | Boolean flag identifying whether queue tracking is disabled or enabled |
| RETENTION | The duration for which dequeued messages will be retained in the queue (TRANSIENT if messages retained for the duration specified in DURATION; PERMANENT if message retained permanently |
| DURATION | Number of days for which the message is retained |
| QUEUE_COMMENT | User comment about the queue |

To enable users to specify a default sorting order for each queue table when a queue table is created, the queuing system also includes a Queue Table Sort table 214. Each row in the Queue Table Sort table 214 represents a sort order for a queue table. The table has the following columns:

| Column | Contents |
|---|---|
| OBJ_NO | Object number |
| SORT_POS | Position of this column; position defines where to order |
| SORT_COLUMN | Position of the column used in the sort expression of DEQUEUE |
| SORT_ORDER | Column order (allowed values are 1 for Ascending, 2 for Descending) |
| COLUMN_NAME | Column name for DEQUEUE sort order |

(Col. 6, l. 45 to col. 8, l. 60).

However, Chandra does not disclose "a table separated from said information queue comprising one or more table records, each said table record comprising an identification of said information in an information queue record, each said table record further comprising a consumer identification field comprising an identification of one of said one or more consumers" as recited in claim 13, as amended. Therefore, Applicants respectfully submit that claim 13, as amended, is patentable over Chandra. Given that claims 14-20 and new claim 27 depend from claim 13, it is respectfully submitted that those claims are also patentable over Chandra.

Claims 21-22

Claims 21-22 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Chandra in view of Rachelson.

The Office action states:

> Chandra discloses a message queue comprising one or more message queue records, each said one or more message queue records comprising a message and a message identification, (Col. 7, lines 6-48); and a history table comprising one or more history records, each of said one or more history record comprising a message identification, a consumer

identification, (Col. 7, lines 6-48); and a work list table comprising one or more work list entries, each said work list entry comprising a message identification, (Col 27, lines 65 – Col. 28, lines 65 and Col. 7, lines 6-48).

(Pg. 10 of October 9, 2002 Office action).

The first cited passage of Chandra discloses:

Each queue table 200 can contain multiple queues 202, 204 each having multiple queue messages 208. In one embodiment, a queue table 200 is structured as shown in FIG. 2. Each row of the queue table 200 represents a message 208 in a queue 202, 204. Some of the columns in a queue table 200 are meta-data describing a queue 202, 204. In one embodiment, each row of the queue table 200 has the following columns:

| Column | Contents |
|---|---|
| QUEUE | Queue name |
| MSG_ID | Message identifier |
| CORR_ID | User-provided correlation identifier |
| MSG_PRIORITY | Message priority |
| MSG_STATE | State of the message (READY to be processed, DELAYED, PROCESSED, or EXPIRED |
| DELAY | Time after which the message will be ready to be processed |
| EXPIRATION | Message expiration time in seconds |
| TIME_MANAGER_INFO | Date used by time manager process to monitor messages |
| LOCAL_ORDER_NO | Local order number of messages |
| CHAIN_NO | Chain number of message |
| DSCN | Dependent transaction number |
| CSCN | Commit transaction number |
| ENQ_TIME | Original enqueue time |
| ENQ_USER_ID | User identifier for the user who enqueued the message |
| ENQ_TXN_ID | Current transaction identifier |
| DEQ_TIME | Time when the message was dequeued |
| DEQ_TXN_ID | Transaction which performed the dequeue |
| DEQ_USER_ID | User id of the user who dequeued the message |
| RETRY_COUNT | Number of retries |
| REPLY_QUEUE_OWNER | Reply queue schema |
| REPLY_QUEUE | Reply queue name |
| EXCEPTION_QUEUE_OWNER | Exception queue schema |
| EXCEPTION_QUEUE | Exception queue name |
| ARG_COUNT | Number of arguments |
| USER_DATA | User data for use by an application or process |

(Col. 7, ll. 6-48).

The second cited passage of Chandra discloses:

As shown in FIG. 4B, a queue table 220 can contain a queue 222 comprising a plurality of messages 224a, 224b, 224c, 224d, 224e. Each of the messages 224a-224e includes a Delay parameter value 234a-234e and a Priority value 236a-236e. The messages 224a-224e are ordered in the queue such that the message 224a with the highest Priority value (100) and the lowest Delay value (0) is at the top. Successive messages have increasingly higher Delay values and increasingly lower Priority values. Each message can have other parameters 250a-250e. The system receives an ENQUEUE request for another message 238 with the Sequence Deviation parameter set to BEFORE and the Relative Message Identifier set to Message 5 (224e). The Delay value 234f of message 238 is 10 and its Priority value 236f is 80. The ENQUEUE request will succeed because Delay value 234f is less than Delay value 234e and Priority value 236f is greater than Priority value 236e.

Time Manager Process

The queuing system further includes a Time Manager process which uses the Delay and Expiration parameter values of an ENQUEUE request to time message delays and notify the DEQUEUE process that a delay is complete. Other functions of the Time Manager process include deleting expired messages from the Queue Table 200 and moving expired messages to an Exception Queue 206. In one embodiment, the Time Manager process communicates with the Timetable database table 216, and is configured as a background process running on the same computer, server, machine or process that is running the queuing system. The Time Manager process uses existing inter-process communications facilities in the database server 310 kernel to communicate with other processes such as ENQUEUE. When the queuing system is started, the Time Manager process is initialized by creating the Timetable 216, creating an index for the Timetable 216 based upon its Time column, and entering a row identifying the Timetable 216 in the Queue Table 200. Preferably initialization of the Time Manager process is triggered by placing a parameter aq.sub.-- tm.sub.-- processes in an initialization parameter file of the database system with which the queue system is integrated. After initialization, the time manager process can be started and stopped using procedures called START.sub.-- TIME.sub.-- MANAGER and STOP.sub.-- TIME.sub.-- MANAGER.

As described herein, the ENQUEUE process writes rows into the Timetable 216 when the Delay or Expiration parameters of an enqueued message 208 contain values.

FIG. 6B, is a flow diagram of Delay and Expiration processing. In step 620, a determination is made as to whether the Delay and Expiration parameters of an enqueued message 208 have values that are greater than zero. If the Delay and Expiration parameters have values that are greater than zero, then at step 622 the ENQUEUE process inserts the referenced message into the Timetable 216. At step 624, it is determined if the value of the SGA variable in the ENQUEUE request is greater than the values of the Delay and Expiration parameters of the enqueued message 208. If the value of the SGA variable in the ENQUEUE request is greater than the values of the Delay and Expiration parameters, then at step 626, the ENQUEUE process posts the time manager, when the Time Manager process is started, in step 700 it reads rows in the Timetable 216 and in step 702 tests whether the Timetable is empty. If so, then control returns to step 700 and the Time Manager essentially goes to sleep until ENQUEUE writes a row into the Timetable. If a Timetable row is found, in step 704 the

> Time Manager reads all entries in the Timetable to locate the message row with the least Time value. In step 706, the Time Manager sets an internal timer or alarm variable to such least Time value. In step 708, the Time Manager tests whether the alarm value is equal to the system clock time, i.e., whether the Time Manager should wake up and start substantive processing.

(Col. 27, l. 65 to col. 29, l. 3).

However, Chandra does not disclose "a work list table separated from said message queue and said history table comprising one or more work list entries, each said work list entry comprising a message identification" as recited in claim 21, as amended.

The Office action further states:

> Chandra does not clearly teach, "a message state identification."
> However Rachelson shows a message state identification, (Col. 9, lines 8-35).

(Pgs. 10-11 of October 9, 2002 Office action).

The cited passage of Rachelson discloses:

> FIG. 9(b) shows an example data structure for storing messages received by a user. Each message is assigned a message id and has one or more flags indicating whether the message has been read by the user and whether the user has archived the message.
>
> If the user selects "List past messages," the user is prompted to determine which past messages the user wishes to view (by month and year) and the system prepares pages containing a printed list (sender, date, messages id, subject, and number of pages) of the messages specified by the user. Control then passes to step C of FIG. 6, where the prepared pages are faxed to the user at the specified fax machine. A preferred embodiment of the present invention also allows the user to retrieve listings of past messages sent/received from a specific person in the user's address book.
>
> If the user selects "Retrieve past messages," the user is prompted to specify which past messages the user wishes to view (by message id) and the system prepares pages containing of the contents of the specified messages. Control then passes to step C of FIG. 6, where the prepared pages are faxed to the user at the specified fax machine.
>
> If the user selects "Forward past messages to someone in address book," the user is prompted to specify which past messages the user wishes to send and to which internet fax number he wishes to send them. The system then prepares pages containing of the contents of the specified messages and the prepared pages are e-mailed to the specified recipient.

(Col. 9, ll. 8-35).

However, Rachelson does not disclose "a work list table separated from said message queue and said history table comprising one or more work list entries, each said work list entry comprising a message identification" as recited in claim 21, as amended.

Even if Chandra and Rachelson were combined, the combination would neither teach nor suggest "a work list table separated from said message queue and said history table comprising one or more work list entries, each said work list entry comprising a message identification" as recited in claim 21, as amended. Accordingly, Applicants respectfully submit that claim 21 is patentable over Chandra in view of Rachelson. Given that claim 22 depends from claim 21, it is respectfully submitted that claim 22 is also patentable over Chandra in view of Rachelson.

Claims 23-26

Claims 23-26 have been allowed. The Office action states:

> [T]he prior art of record, single or in combination, does not teach or fairly suggest the particular step "indicating in a second location that said first consumer has accessed said first piece of information, and indicating in a third location that said second consumer has accessed said first piece of information."

(Pg. 2 of October 9, 2002 Office action).

## CONCLUSION

On the basis of the above remarks, reconsideration and allowance of the claims is believed to be warranted and such action is respectfully requested. If the Examiner has any questions or comments, the Examiner is respectfully urged to contact the undersigned at the number listed below.

Respectfully submitted,

Bingham McCutchen LLP

Dated: 1-9-03

By: _____

Erin C. Ming
Reg. No. 47,797

Three Embarcadero Center, Suite 1800
San Francisco, CA 94111-4067
Telephone: (650) 849-4904
Telefax: (650) 849-4800

## VERSION WITH MARKINGS TO SHOW CHANGES

In the Claims

Claims 13 and 21 have been amended as follows.

13. (Amended) A system for the delivery of information to multiple consumers, said system comprising:

an information queue comprising one or more information queue records, each said information queue record comprising information to be accessed by one or more consumers; and

a table separated from said information queue comprising one or more table records, each said table record comprising an identification of said information in an information queue record, each said table record further comprising a consumer identification field comprising an identification of one of said one or more consumers.

21. (Amended) A system for the delivery of messages to multiple consumers, said system comprising:

a message queue comprising one or more message queue records, each said one or more message queue records comprising a message and a message identification;

a history table separated from said message queue comprising one or more history records, each of said one or more history records comprising a message identification, a consumer identification and a message state identification; and

a work list table separated from said message queue and said history table comprising one or more work list entries, each said work list entry comprising a message identification.